IPv6 OS Fingerprinting Methods: Review

Omar E. Elejla[1(✉)], Bahari Belaton[1], Mohammed Anbar[2], and Basem O. Alijla[3]

[1] School of Computer Science, Universiti Sains Malaysia, Gelugor, Penang, Malaysia
oeoel4_com063@student.usm.my, bahari@usm.my
[2] National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Gelugor, Penang, Malaysia
anbar@nav6.usm.my
[3] Faculty of Information Technology, Islamic University of Gaza, Gaza City, Gaza Strip, Palestine, Israel
balijla@iugaza.edu.ps

Abstract. IPv6 is the new communication protocol which will eventually replace IPv4 is suffering from different security issues. As an initial step to understand IPv6 networks and their vulnerabilities it is of critical importance to identify the characteristics of the connected devices. Detecting the OS fingerprints of these devices is one of these characteristics that are essential to identifying the vulnerabilities of each of them. Currently, few OS detection methods have supported IPv6 protocol, as it did not fully replace IPv4 yet. This paper attempts to describe the existing methods of OS fingerprinting with IPv6, as well as their challenges and limitations. Moreover, this paper studies the available datasets that might be used for IPv6 OS fingerprinting. By understanding the existing methods and datasets, the reader can figure out the current needs for proposing new OS fingerprinting methods for IPv6 protocol.

Keywords: OS fingerprinting · IPv6 protocol · Network security

1 Introduction

IPv6 has been designed to replace IPv4 after IPv4 was criticized in terms of the addresses pool exhaustion and security issues. IPv6 has a four times longer header than IPv4, which can provide an address to every single device in the world. The world's devices count is expected to be 40.9 billion in 2020 [1]. The number of the IPv6 users is continuously increasing on a daily basis. An example of this is presented in Fig. 1 that illustrates the number of Google users who are currently using the IPv6 protocol. IPv6 has built-in security mechanisms such as IPSec protocol which serves to overcome some of IPv4 security issues. In addition, IPv6 introduced address auto configuration and mobility features for the nodes. Another main change in IPv6 compared to IPv4 is its major dependency on the ICMPv6 protocol, where it was optional in IPv4. These changes made the applicability of using IPv4 systems on IPv6 impossible [2].

Despite the security improvements of IPv6 over IPv4, it is still suffering from a set of attacks that exposes its reliability. Recently, several studies showed that IPv6 is

H. Badioze Zaman et al. (Eds.): IVIC 2017, LNCS 10645, pp. 661–668, 2017.
https://doi.org/10.1007/978-3-319-70010-6_61

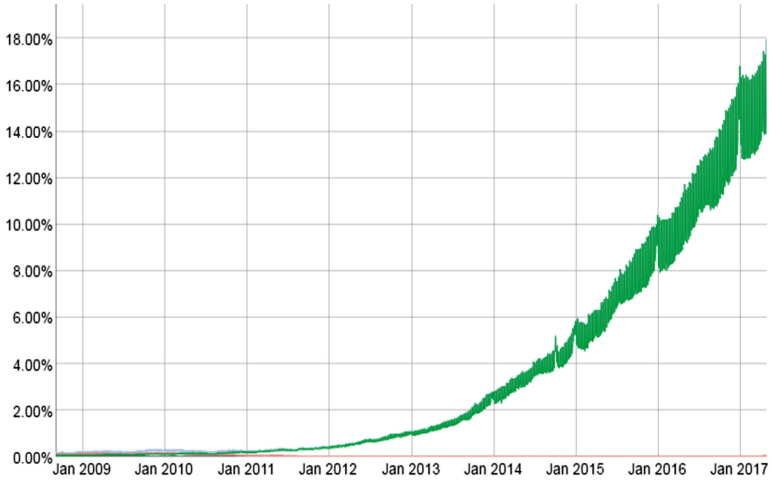

Fig. 1. Google IPv6 users

vulnerable to several types of attacks, which prevented it from being worthy to implement in real networks. IPv6 attacks are categized into two classes, which are either inherited from IPv4 protocol (performed in the same way of IPv4 attacks), or new attacks that depend on the new features of IPv6 [3]. Moreover, some of the IPv6 new features such as the multicast address contribute to making IPv6 attacks easy to perform. An example of this is Denial of Service (DoS) attacks.

Identifying the OS fingerprint is a technique that collects information from a network to determine the number of different hosts connected and the used OSes in the network. Determining the used OSes in the network helps the administrator to realize the security level of the network and find out potential vulnerabilities that the nodes are exposed to. Moreover, the OS fingerprinting detection provides the administrator with information about the unpatched or unauthorized and rogue devices that are attached to the network [4]. As IPv6 networks are vulnerable to many attacks, discovering the used OSes in the networks is a helpful step to improve its security and privacy.

This paper presents a review of the existing methods that are able to identify the fingerprints of the OSes based on IPv6 traffic, as well as studying the available IPv6 datasets for such use. To the best of the author's knowledge, this is the first paper that studied these methods in the light of IPv6 protocol. This paper aims to help in securing IPv6 by presenting this summarized review to the interesting readers for a faster understanding of the OS fingerprinting area. Moreover, having such review opens several questions about the existing methods and their worthiness to be applied to a real network.

OS fingerprinting methods are categorized based on the used technique of collecting the information into two main categories, which are active technique and passive technique. Active techniques depend on sending craft packets to the OSes and identify them based on their responses. Passive techniques prefer to be silent and depend on the normally generated traffics from the OSes. Passive techniques have the

advantage over the active techniques that they do not affect the network performance as they do not generate any traffic (overhead) on the network. In addition, active techniques are never able to determine OSes that are located behind a firewall that crafted packets are unable to reach [5].

This paper is organized as follows: Sect. 2 details some of the existing OS fingerprinting methods and highlights some interesting point of their strength and weakness. Section 3 describes the availability of IPv6 datasets for OS fingerprinting purposes. Section 4 concludes the paper findings with opportunity for future promising technique.

2 IPv6 OS Fingerprinting Methods

IPv4 has been sufficiently studied in the light of OS fingerprinting, and several tools have been proposed for that purpose such as ETTERCAP [6] and Xprobe [7]. IPv4 addresses exhaustion problem is not the only addressed problem by IPv6. Many other features have been either improved, changed or added in the new protocol. Therefore, IPv4 OS fingerprinting methods are unsuitable to be directly applied to IPv6 networks, due to the major changes between the two protocols. However, some researchers have tried to adapt these methods to support IPv6 protocol by considering its new features and fields.

Security researchers realized that IPv6 needs more improvements to reach the goal of securing its networks. Therefore, several OS fingerprinting methods have supported IPv6 protocol as a step towards that goal. These methods are either IPv4 tools that were adapted to support IPv6 protocol by making use of its characteristics, or newly proposed IPv6 OS fingerprinting methods. All these methods are classified based on their information source into active or passive techniques. Figure 2 shows the taxonomy of the existing IPv6 OS fingerprinting methods.

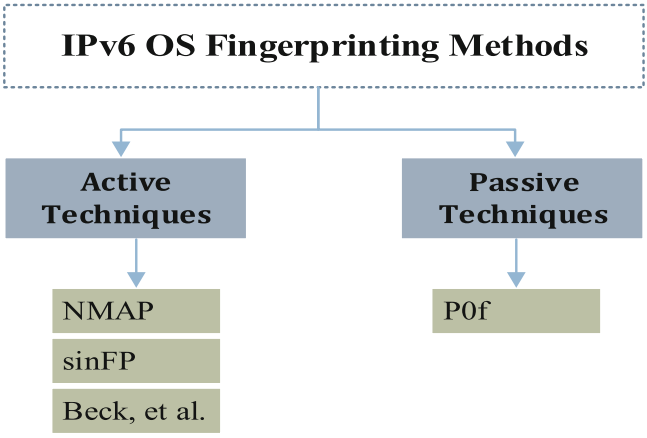

Fig. 2. Taxonomy of IPv6 OS fingerprinting existing methods

2.1 Active Techniques

Active fingerprinting is an aggressive method that probes the targeted devices by sending crafted packets to response with certain messages. The advantages of using such techniques is that they allow their systems to be located at any point of the network, as well as that system can learn more about the network compared to passive techniques. However, active techniques add overhead to the network because of the sent probe packets to the nodes. Moreover, IDSs might identify these packets as malicious behavior that leads to blocking or dropping them.

Several active OS fingerprinting methods have been adapted from IPv4 existing methods or exclusively proposed to identify OSes based on IPv6 traffic.

NMAP [8] is one of the most common free and open source tools that help in exploring and securing networks. It was released in 1998 to provide different services, including port scanning and active OS fingerprinting detection. NMAP uses raw IP packets in novel ways to determine the OSes. NMAP improved its OS detection accuracy compared to other tools by increasing the number of tests (probe packets) that are sent toward the targeted device (currently, 18 packets (TCP, UDP, ICMP) are sent). The devices' responses to the probe packets are compared to the NMAP database of OSes signatures and the closest match is chosen. IPv6 is started to be supported by NMAP since 2011.

The multiple tests that are used in NMAP have improved its detection ability to determine diverse types of OSes. However, these tests exposed the NMAP to be detected as an intrusion by IDS systems due to their suspicious behavior [9]. Moreover, the sent packet might lead to slowing down the performance and affecting the availability of the network. Despite, NMAP understands IPv6 traffic, and its IPv6 database is still considered small, and is unable to automatically determine the OSes based on IPv6 traffic [4, 10]. Although, NMAP has three scanning techniques for IPv6 protocol, practically, TCP scanning is the only working scan [10].

sinFP [11] was released by Patrice Auffret in 2006 mainly to overcome the problems of NMAP tool. sinFP has the ability to use both active and passive fingerprinting techniques using a real SQLite database. In addition to IPv6 support, sinFP has the features of using a few probe packets (3 packets), apply the heuristic matching algorithm, and works in online and offline modes. Another advantage of sinFP is the ability to share the database to be utilized and integrated into other systems.

SinFP supported IPv6 by replacing its equivalent fields to IPv4 such as Identification (ID), Time to Live (TTL), and Don't Fragment (DF) with Flow Label, Hop Limit, and Traffic Class respectively. sinFP is one of the OSes detection tools that support IPv6 OS fingerprinting with its first versions. As SinFP depends on a database of signatures, it suffers from inaccurate detection of OSes that are not pre-recorded in the databases, or when the databases are not up-to-date. Also, sinFP has been criticized for the OS detection being inaccurate, as well as for the potential that the probe packets could be categorized as attacks by IDSs [12]. Therefore, it could not be considered as a reliable tool, and further improvements are still required.

Beck et al. [12] have proposed active OS detection methods exclusively for IPv6 traffics after they concluded that passive techniques are infeasible on IPv6. A simple tool namely osfinger6 has been developed using Python language and Scapy6 packet

manipulation tool [13] to generate the required tests (probe packets). The authors conducted an experiment of sending 156 forged NS messages on a small testbed (6 OSes) using the osfinger6 tool. Based on the observations of the OSes responses, decision tree of the available OSes has been built.

The authors have figured out some surprising results about the OSes' responses to the sent NDP probe packets. However, applying such methods in real networks is irrational due to the same problem of NAMP tool of using a vast number of probe packets. These packets contribute in consuming the networks bandwidth and the possibility of classifying the tool as an intrusion by IDSs. Moreover, the proposed technique has not included the recent changes to the IPv6 extension headers that are defined in RFC 7045 and RFC 6564 [14]. To include the routers to the detected fingerprints, the authors promised to use NS and RS messages.

2.2 Passive Techniques

Passive fingerprinting techniques depend on analyzing the target devices traffics without calling them by sending probe packets. Passive techniques are preferred as the targeted devices are never bothered, thus, the network performance will not be affected. Also, the OS detection tool will be allowed to work normally by the IDSs. The disadvantages of passive techniques are that they could take a long waiting time to get the needed packets from the network to identify the OSes [15]. Moreover, passive techniques cannot work remotely as the tool must be located inside the targeted network to be able to capture the traffic [16].

Despite the advantages of passive OS fingerprinting, it has been applied for IPv6 traffic in one method which is p0f.

P0f (Passive OS Fingerprinting) [17] is the first effective passive OS detection tool. It is one of the most popular passive OS fingerprinting tools and was proposed by Zalewski in 2000. It depends on analyzing the TCP header to extract 9 features (mentioned in [18]) and compare them to a database of signatures to determine the OSes. It also has the ability to detect the OS version, firewall, Network Address Translation (NAT), and the distance to the remote system [12]. P0f does not have user graphical interface, it only can be used through command line prompt.

P0f has the advantage, compared to the active techniques, that it can detect the OSes that are located behind the firewall or NAT. However, the accuracy of identifying the OSes has been criticized as being inaccurate compared to active techniques [10]. In addition, P0f does not work for encrypted traffics where the TCP fields cannot be read. For IPv6 packets, p0f applies the same fingerprints of IPv4 with replacing its fields with their IPv4 equivalents [19]. Despite p0f understanding IPv6 traffic, it does not accurately detect OSes based on it traffics [12].

On summary, few OS fingerprinting methods have been proposed or adapted for IPv6 traffic.

As concluded from Table 1, insufficient research has been conducted on IPv6 OS fingerprinting. This could be due to the lake of implementation of IPv6 protocol on today's networks where most of the OSes still working with IPv4. The existing OS fingerprinting methods were either originally proposed for IPv6, or produced for IPv4 and then adapted to support IPv6. Both have several disadvantages that exposed their

accuracy or affected network performance. Therefore, more work needs to be done to improve the existing methods and overcome their limitations and disadvantages.

Table 1. Summarizes the proposed IPv6 OS detection methods.

Method	Description	Disadvantages
NMAP	• Active OS fingerprinting • 18 probe packets • Understand IPv6 since 2011	• Might be detected as attacker • Might affect the network availability • Small database for IPv6 traffic • Unable to detect IPv6 OSes automatically
sinFP	• Active and passive OS fingerprinting • 3 probe packets • Support IPv6 • Sharable signatures database	• Might be detected as attacker • Might affect the network availability • Inaccurate OS detection
Beck et al. (osfinger6)	• Active OS fingerprinting • 156 probe packets (forged NS) • Support only IPv6 • OSes decision tree was built	• Might be detected as attacker • The probe packets might affect the network availability • Does not include the recent changes to IPv6 extension headers
P0f	• Passive OS fingerprinting • Understand IPv6 • Depends on analyzing 9 TCP features • Detect behind firewall and NAT OSes	• Inaccurate OS detection • Does not work for encrypted traffics • Unable to detect IPv6 OSes automatically

The traditional categorization of OS fingerprinting methods has been used in IPv6 fingerprinting methods. The used techniques have been classified into active and passive techniques. Active techniques have two main problems. One is being exposed to being blocked as an attacker, and the other one is their negative effects on the network. Passive techniques work silently to avoid these problems, and therefore they might be promising to be more applicable in IPv6 OS detection. Despite, the strengths of passive techniques they have been used in one IPv6 fingerprinting tool (p0f) only.

3 Datasets

Several OS fingerprinting methods depend on datasets of traffic to be used for design and evaluation of any new methods. Different IPv4 datasets have been used for this purpose such IRL dataset [20]. On the OS fingerprinting area, these datasets have different purpose which are;

- Understanding the traffic to propose better OS detection methods.
- Choosing the most related features (fingerprints).
- Training different classifiers to discover the most optimal one.
- Evaluating the efficiency of any new proposed method.
- Comparing two different methods by applying them to the same dataset.

In order to propose more IPv6 OS fingerprinting methods, there is an initial need to have a reference dataset with comprehensive OSes. However, to the best of our knowledge, there is a lack of availability of IPv6 datasets for such usage. This could be due to the privacy issues of the IPv6 information (such as IPv6 address and prefix) that might be included in the traffic which might expose the network to outside attacks. However, encryption or mapping techniques can solve such problem. Matoušek et al. [14] is the only research that has noticed this problem and promised to create an IPv6 dataset for OS fingerprinting purpose.

4 Conclusion

Sine IPv6 OS fingerprinting is not widely supported by the security community, this paper opens the door to motivate others to study it. By exploring the existing fingerprinting methods that have the ability to understand and identify OSes based on IPv6 traffic, different points of interest have been highlighted. First, a small number of these methods support IPv6 which either were proposed for IPv6 or adapted from IPv4. Second, these methods are limited by several issues that need to be addressed before their employment on real networks. Third, the lake of the IPv6 datasets is another reason for this shortage of the IPv6 OS fingerprinting methods. Lastly, passive techniques seem more promising to be used in IPv6 OS detection compared to the active techniques due to their silent style, which saves their tools, as well as network resources.

References

1. ABI Research: The Internet of Things will Drive Wireless Connected Devices to 40.9 Billion in 2020. ABI Research (2014). https://www.abiresearch.com/press/the-internet-of-things-will-drive-wireless-connect/
2. Elejla, O.E., Anbar, M., Belaton, B.: ICMPv6-based DoS and DDoS attacks and defense mechanisms. IETE Tech. Rev. 1–18 (2016). doi:10.1080/02564602.2016.1192964
3. Elejla, O.E., Belaton, B., Anbar, M., Alnajjar, A.: Intrusion detection systems of ICMPv6-based DDoS attacks. Neural Comput. Appl. **28**, 1–12 (2016)
4. Schwartzenberg, J.: Using machine learning techniques for advanced passive operating system fingerprinting. Master thesis, University of Twente (2010)
5. Srisuresh, P., Egevang, K.: Traditional IP network address translator (Traditional NAT) (2000)
6. Ornaghi, A., Valleri, M.: Ettercap (2005). http://ettercap.github.io/ettercap/ (2017)
7. Yarochkin, F., Kydyraliev, M., Arkin, O.: Xprobe project (2014). http://x-probe.org/ (2017)

8. Lyon, G.: Nmap–free security scanner for network exploration & security audits (2009). https://nmap.org/ (2017)
9. Greenwald, L.G., Thomas, T.J.: Toward undetected operating system fingerprinting. WOOT **7**, 1–10 (2007)
10. Stopforth, R.: Techniques and countermeasures of TCP/IP OS fingerprinting on Linux Systems. Thesis, University of KwaZulu-Natal, Durban (2007)
11. Auffret, P.: SinFP, January 2007. http://www.gomor.org/sinfp (2017)
12. Beck, F., Festor, O., Chrisment, I.: IPv6 neighbor discovery protocol based OS fingerprinting, Inria (2007)
13. Biondi, P.: Scapy (2011). http://www.secdev.org/projects/scapy/ (2015)
14. Matoušek, P., Ryšavý, O., Grégr, M., Vymlátil, M.: Towards identification of operating systems from the internet traffic: IPFIX monitoring with fingerprinting and clustering. In: 2014 5th International Conference on Data Communication Networking (DCNET), pp. 1–7. IEEE (2014)
15. Prigent, G., Vichot, F., Harrouet, F.: IpMorph: fingerprinting spoofing unification. J. Comput. Virol. **6**(4), 329–342 (2010)
16. Nerakis, E.: IPv6 host fingerprint. Master DTIC Document, Naval Postgraduate School (2006)
17. Zalewski, M.: P0f: Passive OS Fingerprinting Tool (2006). http://lcamtuf.coredump.cx/p0f3/ (2017)
18. Jajodia, S., Subrahmanian, V.S., Swarup, V., Wang, C.: Cyber Deception: Building the Scientific Foundation. Springer International Publishing, Switzerland (2016). doi:10.1007/978-3-319-32699-3
19. Fifield, D., Geana, A., MartinGarcia, L., Morbitzer, M., Tygar, J.D.: Remote operating system classification over IPv6. In: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, pp. 57–67. ACM (2015)
20. IRL Fingerprinting Dataset (2014). http://irl.cs.tamu.edu/projects/sampling/ (2017)